

# HACKATHON 8NETT

A **AgroSolutions** é uma cooperativa agrícola tradicional que busca se modernizar para enfrentar os desafios do século XXI: otimização de recursos hídricos, aumento da produtividade e sustentabilidade. Atualmente, a tomada de decisão no campo é baseada majoritariamente na experiência dos agricultores, sem um forte apoio de dados em tempo real, o que leva a desperdícios e a uma produtividade abaixo do potencial.

Com a visão de implementar a agricultura 4.0, a AgroSolutions decidiu construir uma plataforma de IoT (Internet of Things) e análise de dados para oferecer aos seus cooperados um serviço de agricultura de precisão.

Para isso, a AgroSolutions contratou os alunos do curso **8NETT** para realizar a análise do projeto, arquitetura do software e desenvolvimento do **MVP** desta plataforma inovadora.

## Requisitos Funcionais

- **Autenticação do Usuário (Produtor Rural):**
  - O sistema deve permitir que o produtor rural faça login com e-mail e senha.
- **Cadastro de Propriedade e Talhões:**
  - O produtor deve poder cadastrar sua propriedade e delimitar seus talhões (áreas de plantio), informando a cultura plantada em cada um.
  - Uma propriedade pode ter mais de um talhão.
- **Ingestão de Dados de Sensores (Simulado):**
  - O sistema deve expor uma API para receber dados de sensores de campo (simulados), como umidade do solo, temperatura e nível de precipitação para um determinado talhão.
  - O sistema deve garantir que a API use autenticação segura, JWT ou equivalente.
  - Os dados de sensores podem ser pegos de outras fontes da internet.
- **Dashboard de Monitoramento:**

- O sistema deve permitir que o produtor visualize os dados históricos dos sensores em um gráfico.
- O sistema deve exibir um status geral para cada talhão (ex: "Normal", "Alerta de Seca", "Risco de Praga").
- O dashboard pode ficar no APM para visualização, ou expor em outro lugar que facilite a visualização do Agricultor.
- **Motor de Alertas Simples:**
  - O sistema deve processar os dados recebidos dos sensores e gerar alertas.
  - **Exemplo de regra:** Se a umidade do solo de um talhão ficar abaixo de 30% por mais de 24 horas, gerar um "Alerta de Seca".
  - Os alertas devem ser exibidos no dashboard para o produtor.

### **Requisitos Técnicos Obrigatórios**

A solução deve obrigatoriamente contemplar os seguintes aspectos técnicos:

- **Arquitetura baseada em Microserviços** (ex: Serviço de Identidade, Serviço de Propriedades, Serviço de Ingestão de Dados, Serviço de Análise/Alertas).
- **Orquestração com Kubernetes** – Local (minikube/kind) ou na nuvem (AWS, Azure, GCP).
- **Observabilidade** utilizando APM de sua escolha.
- **Implementação de mensageria** (ex: RabbitMQ, Kafka) para a comunicação entre os microserviços, especialmente para a ingestão e processamento de dados dos sensores.
- **Pipeline de CI/CD automatizado**, utilizando GitHub Actions, Azure DevOps ou equivalente.
- **Adoção das melhores práticas** de arquitetura de software.

### **Requisitos Técnicos Opcionais (Bônus)**

Os alunos podem incluir um ou mais dos seguintes requisitos opcionais **(Não valerá nota, só acrescentará um desafio a mais e poderá ser usado para desempate para o ganhador do prêmio):**

- **Uso de banco de dados NoSQL** como MongoDB ou InfluxDB (ideal para séries temporais) para armazenar os dados dos sensores.
- **Componentes Serverless**, com uso de API Gateway (AWS, Azure ou Kong) e Functions (AWS Lambda ou Azure Functions) para o microserviço de ingestão de dados.
- **Integração com uma API de previsão do tempo** para exibir informações climáticas no dashboard.

## **Entregáveis Mínimos**

Os grupos deverão entregar os seguintes itens:

### **1. Desenho da Solução MVP**

- Diagrama da arquitetura da solução.
- Justificativa técnica das decisões arquiteturais.

### **2. Demonstração da Infraestrutura**

- Aplicação rodando em ambiente de nuvem ou local.
- Evidências de uso de Kubernetes e APM (Métricas, Traces e Logs).
- Evidências de dashboard de monitoramento e alertas (alertas customizados como, Alertas de seca, Risco de pragas etc.).

### **3. Demonstração da Esteira de CI/CD**

- Explicação e demonstração do pipeline de deploy.
- Não é necessário rodar a pipeline, basta mostrar os checks verdes garantindo que tudo rodou corretamente.
- Se a opção escolhida para execução de deploy seja local, será obrigatório ao menos testes unitários, build da imagem e envio para algum registry (ex: dockerhub).

### **4. Demonstração do MVP**

- A aplicação funcional deve contemplar os seguintes recursos:
  - Autenticação do Produtor Rural.
  - Cadastro de Propriedade/Talhão.
  - Envio de dados de sensores (via API de simulação).

- Visualização dos dados e alertas no Dashboard.

**5. Gravação do vídeo de demonstração** dos requisitos/entregáveis mínimos

- Explicação e demonstração geral dos itens 1, 2, 3 e 4.
- O vídeo deve ter no máximo 15 minutos, se passar de 15 minutos poderá sofrer diminuição na pontuação deste requisito.

**6. Compartilhar link do código fonte.**

- Para que a avaliação seja feita da forma correta, os professores vão avaliar o repositório com o conteúdo entregue do Hackaton.
- De preferência, os repositórios devem estar públicos para que no momento da correção, os professores não fiquem travados.

**7. Relatório de entrega (PDF ou TXT) – esse arquivo deve ser postado na data da entrega, contendo:**

- Nome do grupo.
- Participantes e usernames no Discord.
- Link da documentação.
- Link do(s) repositório(s).
- Link do vídeo salvo no Youtube ou lugar de sua preferência.

Lembramos que caso você tenha qualquer dúvida, é só nos chamar no Discord!